Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. **(Currently Amended)** A computer implemented method for enriching an input network with knowledge from a fractal semantic knowledge network, wherein said input network comprises objects and pointers between said objects, and the knowledge network comprises semantic units and a plurality of modules arranged in a fractal semantic network, whereby any one of said modules is associated with one or more of said semantic units such that the respective module is able to operate on the one or more of said semantic units, said method comprising the steps of:

a. finding a counterpart element for an object or a pointer by looking for a semantic unit that is related to the object or the pointer;

b. establishing a classification connection between the object or the pointer and its counterpart element;

c. assigning the module that is associated with the counterpart element, if any, to the object or the pointer;

d. determining the neighborhood of the object or the pointer in the input network and the neighborhood of the counterpart element in the knowledge network, and comparing the neighborhoods to verify the classification connection.

2. **(Original)** The method of claim 1, further comprising the step of:

e. logically segmenting the input network by having another module explore the neighborhoods of semantic units to find an upward neighbor semantic unit or segment on a higher scale in the knowledge network, and by adding a corresponding segment unit to the input network.

3. **(Original)** The method of claim 2, whereby a classification connection is established between the segment unit and the semantic unit or segment which is on a higher scale in the knowledge network.

4. **(Original)** The method of claim 1, whereby steps a) through d) are repeated iteratively until a classification connection is found for some or all of the objects and/or pointers.

5. **(Original)** The method of claim 2, whereby steps a) through e) are repeated iteratively until classification connections are found for some or all of the objects and/or pointers of the input network and some or all of the objects and/or pointers of the input network are segmented on higher levels of hierarchy.

6. **(Original)** The method of claim 1 wherein the modules are capable of operating independently and concurrently.

7. **(Original)** The method of claim 1 wherein some of the semantic units comprise connections that are hierarchical connections being employed to group some of

the semantic units together either by similarity with other semantic units or by functionality with other semantic units.

8. **(Original)** The method of claim 1 whereby more than one module can be associated with a semantic unit, or can be assigned to an object or pointer.

9. **(Original)** The method of claim 1 wherein the knowledge network comprises a root semantic unit.

10. **(Original)** The method of claim 9, wherein the root semantic unit is considered to be a counterpart element for an object for which no counterpart element is found in step a).

11. **(Original)** The method of claim 1 whereby a search algorithm is employed in step a) for finding a counterpart element.

12. **(Original)** The method of claim 11, wherein the search algorithm is a string-matching algorithm, a partial string-matching algorithm, a fuzzy string-matching algorithm, an algorithm that analyzes word characteristics, or an algorithm that analyzes frequency spectra.

13. **(Original)** The method of claim 1, whereby step a) is only carried out for certain groups of objects, preferably verbs and/or nouns.

14. **(Original)** The method of claim 1, wherein the module is a software object.

15. **(Original)** The method of claim 14, whereby in step c) the module is assigned to the object or pointer by copying or cloning the software object.

16. **(Original)** The method of claim 1 whereby the modules can be individually triggered to initiate any of the steps b), c), d), or e).

17. **(Original)** The method of claim 16, whereby scheduling mechanisms are employed to individually trigger the modules.

18. **(Original)** The method of claim 17, whereby there is one scheduling mechanism which ensures that those modules which are assigned to objects representing a verb are triggered first.

19. **(Original)** The method of claim 17, whereby the scheduling mechanisms are employed to ensure that the modules are triggered in a predefined order.

20. **(Original)** The method of claim 1 whereby the module after having been assigned to an object or pointer still remembers to which semantic unit it was associated.

21. **(Original)** The method of claim 1 whereby during step d) or e) the module verifies whether the neighborhood of the object or pointer to which it is assigned resembles the neighborhood of the semantic unit to which it is associated.

22. **(Original)** The method of claim 1, whereby connections are established between objects and/or pointers of the input network and semantic units of the knowledge network.

23. **(Original)** The method of claim 1, whereby a classification probability (Cx) is assigned to the classification connection such that a high classification probability (Cx) indicates a good match between an object or a pointer and a counterpart element.

24. **(Original)** The method of claim 23, whereby the classification probability (Cx) is increased if the module finds out that the neighborhood of the object or the pointer to which the module is assigned resembles the neighborhood of the semantic unit to which the module is associated.

25. **(Currently Amended)** A computer software module being assignable to an object or a pointer of an input network, whereby said software module when being triggered, performs the steps of :

comparing the neighborhood of the object or pointer to which it is assigned with a neighborhood of a counterpart element of a fractal semantic knowledge network in order to verify a classification connection between said object or pointer to which it is assigned and the counterpart element;

updating a classification value (Cx) of the classification connection if the neighborhood of said object or pointer in the input network resembles the neighborhood of the counterpart element in the knowledge network.

26. **(Original)** The software module of claim 25, whereby, in comparing the neighborhoods, the software module verifies whether

the object or the pointer in the input network and the counterpart element in the knowledge network are alike; or

the object or the pointer in the input network and the counterpart element in the knowledge network are able to perform similar tasks; or

the object or the pointer in the input network and the counterpart element in the knowledge network have similar goals; or

the object or the pointer in the input network and the counterpart element in the knowledge network is more general or more specific; or

the object or the pointer in the input network and the counterpart element in the knowledge network are constituents or groups; or

the object or the pointer in the input network and the counterpart element in the knowledge network are in similar states.

27. **(Original)** The software module of claim 25, wherein the classification value (Cx) represents the degree of similarity of the neighborhood of the object or pointer and the neighborhood of the counterpart element in the knowledge network.

28. **(Original)** The software module of claim 25 serving as a Classification Janus.

29. **(Original)** A natural language processing system comprising a plurality of software modules pursuant to claim 25.

30. **(Currently Amended)** A computer software module being assignable to an object or a pointer of an input network, whereby said software module when being triggered, performs the steps of:

looking for semantic units in a fractal semantic knowledge database which belong to a counterpart segment on a higher level of hierarchy; and if such a counterpart segment is identified,

creating a new segment unit in the input network, and

creating a classification connection from the new segment unit to the counterpart segment.

31. **(Original)** The software module of claim 30, wherein the software module is designed to allow the

segmentation of input semantic units; or

de-segmentation of input semantic units; or

fusion of two or more input semantic units into one input semantic unit; or

fission of an input semantic unit; or

foundation of segments; or

optimization of a boundary between segments.

32. **(Original)** The software module of claim 30 serving as a Segmentation Janus.

33. **(Original)** A natural language processing system comprising a plurality of software modules pursuant to claim 30.

34. **(Currently Amended)** A natural language processing system implemented in a computer for enriching an input network with knowledge from a fractal semantic knowledge network, whereby the input network comprises objects and pointers between said objects, and the knowledge network comprises semantic units, the system comprising:

a module for finding a counterpart element for an object or a pointer by looking for a semantic unit that is related to the object or the pointer;

a module for establishing a classification connection between the object or the pointer and its counterpart element;

a module for examining the objects' or the pointers' neighborhoods in the input network by comparing them with the counterpart elements' neighborhoods in knowledge network to verify the classification connection.

35. **(Original)** The natural language processing system of claim 34, further comprising a module which logically segments the input network by exploring the neighborhood of semantic units and/or pointers to find an upward neighbor semantic unit

or segment on a higher scale in the knowledge network, and adding corresponding segment units to the input network.

36. **(Original)** The natural language processing system of claim 34 whereby some or all of the modules are software objects.